

Complexity results on w -well-covered graphs

Vadim E. Levit and David Tankus

Department of Computer Science and Mathematics

Ariel University, ISRAEL

{levitv, davidta}@ariel.ac.il

Abstract

A graph G is *well-covered* if all its maximal independent sets are of the same cardinality. Assume that a weight function w is defined on its vertices. Then G is *w -well-covered* if all maximal independent sets are of the same weight. For every graph G , the set of weight functions w such that G is w -well-covered is a *vector space*, denoted $WCW(G)$.

Let B be a complete bipartite induced subgraph of G on vertex sets of bipartition B_X and B_Y . Then B is *generating* if there exists an independent set S such that $S \cup B_X$ and $S \cup B_Y$ are both maximal independent sets of G . A *relating edge* is a generating subgraph in the restricted case that $B = K_{1,1}$.

Deciding whether an input graph G is well-covered is **co-NP**-complete. Therefore finding $WCW(G)$ is **co-NP**-hard. Deciding whether an edge is relating is **co-NP**-complete. Therefore, deciding whether a subgraph is generating is **co-NP**-complete as well.

In this article we discuss the connections among these problems, provide proofs for **NP**-completeness for several restricted cases, and present polynomial characterizations for some other cases.

1 Introduction

1.1 Basic definitions and notation

Throughout this paper G is a simple (i.e., a finite, undirected, loopless and without multiple edges) graph with vertex set $V(G)$ and edge set $E(G)$.

Cycles of k vertices are denoted by C_k . When we say that G does not contain C_k for some $k \geq 3$, we mean that G does not admit subgraphs isomorphic to C_k . It is important to mention that these subgraphs are not necessarily induced. Let $\mathcal{G}(\widehat{C_{i_1}}, \dots, \widehat{C_{i_k}})$ be the family of all graphs which do not contain C_{i_1}, \dots, C_{i_k} .

Let u and v be two vertices in G . The *distance* between u and v , denoted $d(u, v)$, is the length of a shortest path between u and v , where the length of a path is the number of its edges. If S is a non-empty set of vertices, then the *distance* between u and S , denoted $d(u, S)$, is defined by

$$d(u, S) = \min\{d(u, s) : s \in S\}.$$

For every positive integer i , denote

$$N_i(S) = \{x \in V(G) : d(x, S) = i\},$$

and

$$N_i[S] = \{x \in V(G) : d(x, S) \leq i\}.$$

We abbreviate $N_1(S)$ and $N_1[S]$ to be $N(S)$ and $N[S]$, respectively. If S contains a single vertex, v , then we abbreviate $N_i(\{v\})$, $N_i[\{v\}]$, $N(\{v\})$, and $N[\{v\}]$ to be $N_i(v)$, $N_i[v]$, $N(v)$, and $N[v]$, respectively. We denote by $G[S]$ the subgraph of G induced by S . For every two sets, S and T , of vertices of G , we say that S *dominates* T if $T \subseteq N[S]$.

1.2 Well-covered graphs

Let G be a graph. A set of vertices S is *independent* if its elements are pairwise nonadjacent. An independent set of vertices is *maximal* if it is not a subset of another independent set. An independent set of vertices is *maximum* if the graph does not contain an independent set of a higher cardinality.

The graph G is *well-covered* if every maximal independent set is maximum [14]. Assume that a weight function $w : V(G) \rightarrow \mathbb{R}$ is defined on the vertices of G . For every set $S \subseteq V(G)$, define $w(S) = \sum_{s \in S} w(s)$. Then G is *w-well-covered* if all maximal independent sets of G are of the same weight.

The problem of finding a maximum independent set in an input graph is **NP**-complete. However, if the input is restricted to well-covered graphs, then a maximum independent set can be found polynomially using the *greedy algorithm*. Similarly, if a weight function $w : V(G) \rightarrow \mathbb{R}$ is defined on the vertices of G , and G is *w-well-covered*, then finding a maximum weight independent set is a polynomial problem.

The recognition of well-covered graphs is known to be **co-NP**-complete. This was proved independently in [4] and [18]. In [3] it is proven that the problem remains **co-NP**-complete even when the input is restricted to $K_{1,4}$ -free graphs. However, the problem is polynomially solvable for $K_{1,3}$ -free graphs [19, 20], for graphs with girth at least 5 [6], for graphs with a bounded maximal degree [2], for chordal graphs [16], and for graphs without cycles of lengths 4 and 5 [7]. It should be emphasized that the forbidden cycles are not necessarily induced.

For every graph G , the set of weight functions w for which G is *w-well-covered* is a vector space [2]. That vector space is denoted $WCW(G)$ [1]. Clearly, $w \in WCW(G)$ if and only if G is *w-well-covered*. Since recognizing well-covered graphs is **co-NP**-complete, finding the vector space $WCW(G)$ of an input graph G is **co-NP**-hard. However, finding $WCW(G)$ can be done polynomially when the input is restricted to graphs with a bounded maximal degree [2], and to graphs without cycles of lengths 4, 5 and 6 [11].

1.3 Generating subgraphs and relating edges

We use the following notion, which has been introduced in [10]. Let B be an induced complete bipartite subgraph of G on vertex sets of bipartition B_X and B_Y . Assume that there exists an independent set S such that each of $S \cup B_X$ and $S \cup B_Y$ is a maximal independent set of G . Then B is a *generating* subgraph of G , and it *produces* the restriction: $w(B_X) = w(B_Y)$. Every weight function w such that G is w -well-covered must *satisfy* the restriction $w(B_X) = w(B_Y)$. The set S is a *witness* that B is generating.

In the restricted case that the generating subgraph B is isomorphic to $K_{1,1}$, call its vertices x and y . In that case xy is a *relating* edge, and $w(x) = w(y)$ for every weight function w such that G is w -well-covered.

The decision problem whether an edge in an input graph is relating is **NP**-complete [1], and it remains **NP**-complete even when the input is restricted to graphs without cycles of lengths 4 and 5 [12]. Therefore, recognizing generating subgraphs is also **NP**-complete when the input is restricted to graphs without cycles of lengths 4 and 5. However, recognizing relating edges can be done polynomially if the input is restricted to graphs without cycles of lengths 4 and 6 [12], and to graphs without cycles of lengths 5 and 6 [11].

It is also known that recognizing generating subgraphs is a polynomial problem when the input is restricted to graphs without cycles of lengths 4, 6 and 7 [10], to graphs without cycles of lengths 4, 5 and 6 [11], and to graphs without cycles of lengths 5, 6 and 7 [11].

2 Four problems

The subject of this article is the following four problems and the connections among them.

- **WC problem:**
Input: A graph G .
Question: Is G well-covered?
- **WCW problem:**
Input: A graph.
Output: The vector space $WCW(G)$.
- **GS problem:**
Input: A graph G , and an induced complete bipartite subgraph B of G .
Question: Is B generating?
- **RE problem:**
Input: A graph G , and an edge $xy \in E(G)$.
Question: Is xy relating?

If we know the output of the **WCW** problem for an input graph G , then we know the output of the **WC** problem for the same input G : The graph G is well-covered if and only if $w \equiv 1$ belongs to $WCW(G)$. Therefore, the **WC** problem is not harder than the **WCW** problem. Let Ψ be a family of graphs. If the **WCW** problem can be solved polynomially when its input is restricted to Ψ , then also the **WC** problem is polynomial when its input is restricted to Ψ . On the other hand, if the **WC** problem is **co-NP**-complete when its input is restricted to Ψ , then the **WCW** problem is **co-NP**-hard when its input is restricted to Ψ .

A similar connection exists between the **GS** problem and the **RE** problem, since an edge is a restricted case of a bipartite graph. Therefore, for every family Ψ of graphs, if the **GS** problem is polynomial, then the **RE** problem is also polynomial, and if the **RE** problem is **NP**-complete then the **GS** problem is **NP**-complete as well.

It follows from the following lemma that the **GS** problem is not more difficult than the **WCW** problem.

Lemma 1 *Let Ψ be a family of graphs. If the **WCW** problem is polynomial solvable when its input is restricted to Ψ , then also the **GS** problem is polynomial solvable when its input is restricted to Ψ .*

Proof. Let $I = (G, B = (B_X, B_Y))$ be an instance of the **GS** problem, where $G \in \Psi$. The following algorithm decides polynomially whether I is positive. Find polynomially $WCW(G)$. If there exists a weight function $w \in WCW(G)$ such that $w(B_X) \neq w(B_Y)$ then B is not generating, and I is negative. Otherwise, I is positive. ■

3 NP-complete cases

A *binary variable* is a variable whose value is either 0 or 1. If x is a binary variable then its *negation* is denoted by \bar{x} . Each of x and \bar{x} are called *literals*. Let $X = \{x_1, \dots, x_n\}$ be a set of binary variables. A *clause* c over X is a set of literals belonging to $\{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$ such that c does not contain both a variable and its negation. A *truth assignment* is a function

$$\Phi : \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\} \longrightarrow \{0, 1\}$$

such that

$$\Phi(\bar{x}_i) = 1 - \Phi(x_i) \text{ for each } 1 \leq i \leq n.$$

A truth assignment Φ *satisfies* a clause c if c contains at least one literal l such that $\Phi(l) = 1$.

3.1 Relating edges in bipartite graphs

In this subsection we consider the following problems:

- **SAT problem:**

Input: A set X of binary variables and a set C of clauses over X .

Question: Is there a truth assignment for X which satisfies all clauses of C ?

- **USAT problem:**

Input: A set X of binary variables and two sets, C_1 and C_2 , of clauses over X , such that all literals of the clauses belonging to C_1 are variables, and all literals of clauses belonging to C_2 are negations of variables.

Question: Is there a truth assignment for X , which satisfies all clauses of $C = C_1 \cup C_2$?

By Cook-Levin Theorem, the **SAT** problem is **NP**-complete. We prove that the same holds for the **USAT** problem.

Lemma 2 *The USAT problem is NP-complete.*

Proof. Obviously, the **USAT** problem is **NP**. We prove its **NP**-completeness by showing a reduction from the **SAT** problem. Let

$$I_1 = (X = \{x_1, \dots, x_n\}, C = \{c_1, \dots, c_m\})$$

be an instance of the **SAT** problem. Define $Y = \{x_1, \dots, x_n, y_1, \dots, y_n\}$, where y_1, \dots, y_n are new variables. For every $1 \leq j \leq m$, let c'_j be the clause obtained from c_j by replacing $\overline{x_i}$ with y_i for each $1 \leq i \leq n$. Let $C' = \{c'_1, \dots, c'_m\}$. For each $1 \leq i \leq n$ define two new clauses, $d_i = (x_i, y_i)$ and $e_i = (\overline{x_i}, \overline{y_i})$. Let $D = \{d_1, \dots, d_n\}$ and $E = \{e_1, \dots, e_n\}$. Obviously, all literals of $C' \cup D$ are variables, and all literals of E are negations of variables. Hence, $I_2 = (Y, C' \cup D \cup E)$ is an instance of the **USAT** problem, see Example 3. It remains to prove that I_1 and I_2 are equivalent.

Assume that I_1 is a positive instance of the **SAT** problem. There exists a truth assignment

$$\Phi_1 : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

which satisfies all clauses of C . Extract Φ_1 to a truth assignment

$$\Phi_2 : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}, y_1, \overline{y_1}, \dots, y_n, \overline{y_n}\} \longrightarrow \{0, 1\}$$

by defining $\Phi_2(y_i) = 1 - \Phi_1(x_i)$ for each $1 \leq i \leq n$. Clearly, Φ_2 is a truth assignment which satisfies all clauses of $C' \cup D \cup E$. Hence, I_2 is a positive instance of the **USAT** problem.

Assume I_2 is a positive instance of the **USAT** problem. There exists a truth assignment

$$\Phi_2 : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}, y_1, \overline{y_1}, \dots, y_n, \overline{y_n}\} \longrightarrow \{0, 1\}$$

that satisfies all clauses of $C' \cup D \cup E$. For every $1 \leq i \leq n$ it holds that $\Phi_2(y_i) = 1 - \Phi_2(x_i)$, or otherwise one of d_i and e_i is not satisfied. Therefore, I_1 is a positive instance of the **SAT** problem. ■

Example 3 *The following contains both an instance of the **SAT** problem and an equivalent instance of the **USAT** problem.*

$$\begin{aligned}
I_1 &= (x_1 + \overline{x_2} + x_3) \cdot (x_1 + x_3 + x_4 + x_5) \cdot \\
&\quad (\overline{x_1} + x_2 + \overline{x_3} + x_4) \cdot (x_1 + x_2 + \overline{x_4} + \overline{x_5}), \\
I_2 &= (x_1 + y_2 + x_3) \cdot (x_1 + x_3 + x_4 + x_5) \cdot (y_1 + x_2 + y_3 + x_4) \cdot \\
&\quad (x_1 + x_2 + y_4 + y_5) \cdot (x_1 + y_1) \cdot (\overline{x_1} + \overline{y_1}) \cdot (x_2 + y_2) \cdot (\overline{x_2} + \overline{y_2}) \cdot \\
&\quad (x_3 + y_3) \cdot (\overline{x_3} + \overline{y_3}) \cdot (x_4 + y_4) \cdot (\overline{x_4} + \overline{y_4}) \cdot (x_5 + y_5) \cdot (\overline{x_5} + \overline{y_5}).
\end{aligned}$$

The following theorem is the main result of this section.

Theorem 4 *The **RE** problem is **NP**-complete even if its input is restricted to bipartite graphs.*

Proof. The problem is obviously **NP**. We prove **NP**-completeness by showing a reduction from the **USAT** problem. Let

$$I_1 = (X = \{x_1, \dots, x_n\}, C = C_1 \cup C_2)$$

be an instance of the **USAT** problem, where $C_1 = \{c_1, \dots, c_m\}$ is a set of clauses which contain only variables, and $C_2 = \{c'_1, \dots, c'_{m'}\}$ is a set of clauses which contain only negations of variables. Define a graph B as follows:

$$\begin{aligned}
V(B) &= \{x, y\} \cup \{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\} \cup \\
&\quad \{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\},
\end{aligned}$$

$$\begin{aligned}
E(B) &= \{xy\} \cup \{xv_j : 1 \leq j \leq m\} \cup \{yv'_j : 1 \leq j \leq m'\} \cup \\
&\quad \{v_j u_i : x_i \text{ appears in } c_j\} \cup \{v'_j u'_i : \overline{x_i} \text{ appears in } c'_j\} \cup \{u_i u'_i : 1 \leq i \leq n\}.
\end{aligned}$$

Clearly, B is bipartite, and the vertex sets of its bipartition are

$$\{u_i : 1 \leq i \leq n\} \cup \{x\} \cup \{v'_j : 1 \leq j \leq m'\}$$

and

$$\{v_j : 1 \leq j \leq m\} \cup \{y\} \cup \{u'_i : 1 \leq i \leq n\}.$$

Let $I_2 = (B, xy)$ be an instance of the **RE** problem. It is necessary to prove that I_1 and I_2 are equivalent.

Assume that I_1 is a positive instance of the **USAT** problem. Let

$$\Phi : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

be a truth assignment which satisfies all clauses of C . Let

$$S = \{u_i : \Phi(x_i) = 1\} \cup \{u'_i : \Phi(x_i) = 0\}.$$

Obviously, S is independent. Since Φ satisfies all clauses of C , every vertex of

$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\}$$

is adjacent to a vertex of S . Hence, S is a witness that xy is a relating edge. Therefore, I_2 is a positive instance of the **RE** problem.

On the other hand, assume that I_2 is a positive instance of the **RE** problem. Let S be a witness of xy . Since S is a maximal independent set of

$$\{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\},$$

exactly one of u_i and u'_i belongs to S , for every $1 \leq i \leq n$. Let

$$\Phi : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

be a truth assignment defined by: $\Phi(x_i) = 1 \iff u_i \in S$. The fact that S dominates

$$\{v_j : 1 \leq j \leq m\} \cup \{v'_j : 1 \leq j \leq m'\}$$

implies that all clauses of C are satisfied by Φ . Therefore, I_1 is a positive instance of the **USAT** problem. ■

Example 5 *Let*

$$I_1 = (x_1 + x_2 + x_3) \cdot (x_2 + x_4) \cdot (x_1 + x_4) \cdot (x_1 + x_5 + x_6) \cdot \\ (x_3 + x_5 + x_6) \cdot (\overline{x_1} + \overline{x_2} + \overline{x_3}) \cdot (\overline{x_2} + \overline{x_3} + \overline{x_4} + \overline{x_5}) \cdot (\overline{x_2} + \overline{x_4} + \overline{x_5} + \overline{x_6})$$

*be an instance of the **USAT** problem. Then $I_2 = (G, xy)$ is an equivalent instance of the **RE** problem, where G is the graph shown in Figure 1. The instance I_1 is positive because $x_1 = x_3 = x_4 = x_5 = 1$, $x_2 = x_6 = 0$ is a satisfying assignment. The corresponding witness that I_2 is positive is the set $\{u_1, u'_2, u_3, u_4, u_5, u'_6\}$.*

Theorem 6 *The **GS** problem is **NP**-complete, while the **WCW** problem is **co-NP**-hard, when their inputs are restricted to bipartite graphs.*

Proof. Since a relating edge is an instance of a generating subgraph, Theorem 4 implies that the **GS** problem is **NP**-complete, when its input is restricted to bipartite graphs. Lemma 1 implies that the **WCW** problem is **co-NP**-hard, when its input is restricted to bipartite graphs. ■

3.2 Graphs with girth at least 6

In this subsection we consider the following problems:

- **3-SAT problem:**

Input: A set X of binary variables and a set C of clauses over X such that every clause contains exactly 3 literals.

Question: Is there a truth assignment for X satisfying all clauses of C ?

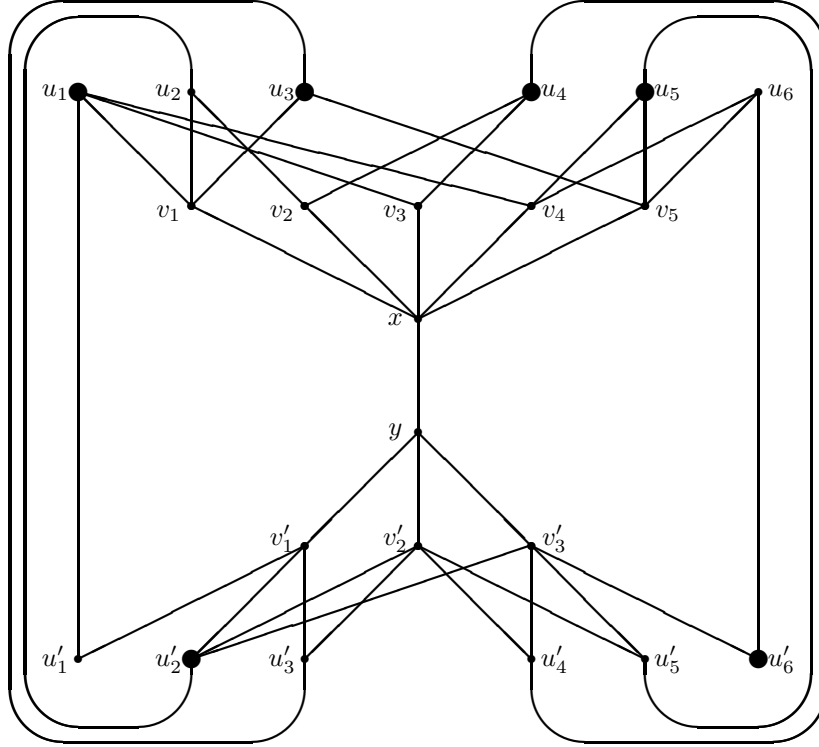


Figure 1: An example of the reduction from the **USAT** problem to the **RE** problem.

• **DSAT problem:**

Input: A set X of binary variables and a set C of clauses over X such that the following holds:

- Every clause contains 2 or 3 literals.
- Every two clauses have at most one literal in common.
- If two clauses, c_1 and c_2 , have a common literal l_1 , then there does not exist a literal l_2 such that c_1 contains l_2 and c_2 contains $\overline{l_2}$.

Question: Is there a truth assignment for X satisfying all clauses of C ?

Let $I = (X, C)$ be an instance of the **3-SAT** problem. A *bad pair of clauses* is a set of two clauses $\{c_1, c_2\} \subseteq C$ such that there exist literals, l_1, l_2, l_3, l_4, l_5 , and:

- $c_1 = (l_1, l_2, l_3)$ and $c_2 = (l_1, l_4, l_5)$;
- either $l_2 = l_4$ or $l_2 = \overline{l_4}$.

Clearly, an instance of the **3-SAT** problem with no bad pair of clauses is also an instance of the **DSAT** problem. The **3-SAT** problem is known to be **NP**-complete. We prove that the same holds for the **DSAT** problem.

Lemma 7 *The **DSAT** problem is **NP**-complete.*

Proof. Obviously, the **DSAT** problem is **NP**. We prove its **NP**-completeness by showing a reduction from the **3-SAT** problem. Let

$$I_1 = (X = \{x_1, \dots, x_n\}, C = \{c_1, \dots, c_m\})$$

be an instance of the **3-SAT** problem.

Assume that there exists a bad pair of clauses, $\{c_{j_1}, c_{j_2}\} \subseteq C$, i.e. there exist literals, l_1, l_2, l_3, l_4, l_5 , such that:

- $c_{j_1} = (l_1, l_2, l_3)$ and $c_{j_2} = (l_1, l_4, l_5)$;
- either $l_2 = l_4$ or $l_2 = \overline{l_4}$.

Define a new binary variable x_{n+1} , and new clauses $c_{j_2}^1 = (l_1, x_{n+1}, l_5)$, $c_{j_2}^2 = (\overline{l_4}, x_{n+1})$, and $c_{j_2}^3 = (l_4, \overline{x_{n+1}})$. Then

$$I'_1 = (X \cup \{x_{n+1}\}, (C \setminus \{c_{j_2}\}) \cup \{c_{j_2}^1, c_{j_2}^2, c_{j_2}^3\})$$

is an instance of the **SAT** problem.

We prove that I_1, I'_1 are equivalent. Assume that I_1 is a positive instance of the **3-SAT** problem. There exists a truth assignment

$$\Phi_1 : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

which satisfies all clauses of C . Extract Φ_1 to a truth assignment

$$\Phi_2 : \{x_1, \overline{x_1}, \dots, x_{n+1}, \overline{x_{n+1}}\} \longrightarrow \{0, 1\}$$

by defining $\Phi_2(x_{n+1}) = \Phi_1(l_4)$. Clearly, Φ_2 satisfies all clauses of I'_1 . On the other hand, assume that there exists a truth assignment

$$\Phi_2 : \{x_1, \overline{x_1}, \dots, x_{n+1}, \overline{x_{n+1}}\} \longrightarrow \{0, 1\}$$

which satisfies all clauses of I'_1 . Clauses $c_{j_2}^2$, and $c_{j_2}^3$ imply that $\Phi_2(x_{n+1}) = \Phi_2(l_4)$. Therefore, I_1 is a positive instance of the **3-SAT** problem.

The new clauses we added contain a new binary variable. Hence, they do not belong to bad pairs of clauses. Moreover, the clause c_{j_2} which belongs to a bad pair in I_1 was omitted in I'_1 . Hence, the number of bad pairs of clauses in I'_1 is smaller than the one in I_1 .

Repeat the process until an instance without bad pairs of clauses is obtained, and denote that instance I_2 . Clearly, every clause of I_2 has 2 or 3 literals. Hence, I_2 is an instance of the **DSAT** problem, and I_1 and I_2 are equivalent. ■

Example 8 The following contains an instance of the **3-SAT** problem and an equivalent instance of the **DSAT** problem.

$$\begin{aligned}
I_1 &= (x_1 + \overline{x_2} + x_3) \cdot (x_1 + x_3 + x_4) \cdot (x_1 + x_3 + x_5) \cdot (\overline{x_3} + \overline{x_4} + x_5) \cdot \\
&\quad (x_2 + \overline{x_3} + \overline{x_4}) \cdot (\overline{x_1} + \overline{x_4} + \overline{x_5}), \\
I_2 &= (x_1 + \overline{x_2} + x_3) \cdot (x_1 + y_3 + x_4) \cdot (x_1 + z_3 + x_5) \cdot (\overline{x_3} + \overline{x_4} + x_5) \cdot \\
&\quad (x_2 + \overline{x_3} + y_4) \cdot (\overline{x_1} + \overline{x_4} + y_5) \cdot (\overline{x_3} + y_3) \cdot (x_3 + \overline{y_3}) \cdot \\
&\quad (\overline{x_3} + z_3) \cdot (x_3 + \overline{z_3}) \cdot (x_4 + y_4) \cdot (\overline{x_4} + \overline{y_4}) \cdot (x_5 + y_5) \cdot (\overline{x_5} + \overline{y_5}).
\end{aligned}$$

Theorem 9 The following problem is **NP-complete**:

Input: A graph $G \in \mathcal{G}(\widehat{C_3}, \widehat{C_4}, \widehat{C_5})$ and an induced complete bipartite subgraph B of G .

Question: Is B generating?

Proof. The problem is obviously **NP**. We prove its **NP-completeness** by showing a reduction from the **DSAT** problem. Let

$$I = (X = \{x_1, \dots, x_n\}, C = \{c_1, \dots, c_m\})$$

be an instance of the **DSAT** problem. Define a graph G as follows.

$$\begin{aligned}
V(G) &= \{y\} \cup \{a_j : 1 \leq j \leq m\} \cup \{v_j : 1 \leq j \leq m\} \cup \\
&\quad \{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\}.
\end{aligned}$$

$$\begin{aligned}
E(G) &= \{ya_j : 1 \leq j \leq m\} \cup \{a_jv_j : 1 \leq j \leq m\} \cup \{v_ju_i : x_i \text{ appears in } c_j\} \cup \\
&\quad \{v_ju'_i : \overline{x_i} \text{ appears in } c_j\} \cup \{u_iu'_i : 1 \leq i \leq n\}.
\end{aligned}$$

Since a clause can not contain both a variable and its negation, G does not contain C_3 . The fact that there are no pairs of bad clauses implies that G does not contain C_4 and C_5 . Hence, $G \in \mathcal{G}(\widehat{C_3}, \widehat{C_4}, \widehat{C_5})$. Let $B = G[\{y\} \cup \{a_j : 1 \leq j \leq m\}]$. Obviously, B is complete bipartite. Let $J = (G, B)$ be an instance of the **GS** problem. It remains to prove that I and J are equivalent.

Assume that I is positive, and let

$$\Phi : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

be a truth assignment which satisfies all clauses of C . Define

$$S = \{u_i : \Phi(x_i) = 1\} \cup \{u'_i : \Phi(x_i) = 0\}.$$

Obviously, S is independent. Since Φ satisfies all clauses of C , the set S dominates $\{v_j : 1 \leq j \leq m\}$. Hence, S is a witness that B is generating, i.e., J is positive.

Assume that J is positive. Let S be a witness that B is generating, and let S^* be a maximal independent set of $\{u_i : 1 \leq i \leq n\} \cup \{u'_i : 1 \leq i \leq n\}$ which contains S . For every $1 \leq i \leq n$, it holds that $|S^* \cap \{u_i, u'_i\}| = 1$. Define

$$\Phi : \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\} \longrightarrow \{0, 1\}$$

by $\Phi(x_i) = 1 \iff u_i \in S^*$ for every $1 \leq i \leq n$. Since S^* dominates $\{v_j : 1 \leq j \leq m\}$, the function Φ satisfies all clauses of C , and I is a positive instance. ■

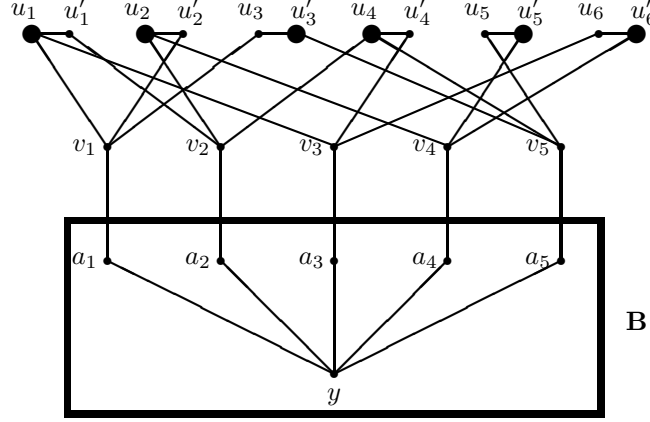


Figure 2: An example of the reduction from the **DSAT** problem to the **GS** problem.

Example 10 *Let*

$$I_1 = (x_1 + \overline{x_2} + x_3) \cdot (\overline{x_1} + x_2 + x_4) \cdot (x_1 + \overline{x_4} + x_6) \cdot (x_2 + \overline{x_5} + \overline{x_6}) \cdot (\overline{x_3} + x_4 + x_5)$$

be an instance of the **DSAT** problem. Then $I_2 = (G, B)$ is an equivalent instance of the **GS** problem, where G and B are the graphs shown in Figure 2. The instance I_1 is positive because $x_1 = x_2 = x_4 = 1$, $x_3 = x_5 = x_6 = 0$ is a satisfying assignment. The corresponding witness that I_2 is positive is the set $\{u_1, u_2, u'_3, u_4, u'_5, u'_6\}$.

It follows from Theorem 9 that the **GS** problem is **NP**-complete when its input is restricted to $\mathcal{G}(\widehat{C_4}, \widehat{C_5})$. However, for that family of graphs even the **RE** problem is **NP**-complete [12].

The following Theorem follows immediately from Theorem 9 and Lemma 1.

Theorem 11 *The **WCW** problem is **co-NP**-hard, when its input is restricted to $\mathcal{G}(\widehat{C_3}, \widehat{C_4}, \widehat{C_5})$.*

It follows from Theorem 11 that the **WCW** problem is **co-NP**-hard, when its input is restricted to $\mathcal{G}(\widehat{C_3}, \widehat{C_4})$ and to $\mathcal{G}(\widehat{C_4}, \widehat{C_5})$. However, the **WC** problem is known to be polynomial for $\mathcal{G}(\widehat{C_3}, \widehat{C_4})$ [6] and for $\mathcal{G}(\widehat{C_4}, \widehat{C_5})$ [7].

3.3 $K_{1,4}$ -free graphs

Theorem 12 [3] *The following problem is **NP**-complete:*

Input: A $K_{1,4}$ -free graph G .

Question: Is G well-covered?

We use Theorem 12 to prove the following.

Theorem 13 *The **GS** problem is **NP**-complete even when its input is restricted to $K_{1,4}$ -free graphs.*

Proof. Let G be a $K_{1,4}$ -free graph. An induced complete bipartite subgraph of G is isomorphic to $K_{i,j}$, for $1 \leq i \leq j \leq 3$. Hence, the number of these subgraphs is $O(n^6)$, which is polynomial. Every unbalanced induced complete bipartite subgraph of G is a copy of $K_{1,2}$ or $K_{1,3}$ or $K_{2,3}$. The number of these subgraphs is $O(n^5)$.

Assume, on the contrary, that there exists a polynomial algorithm solving the **GS** problem for $K_{1,4}$ -free graphs. The following algorithm decides polynomially whether a $K_{1,4}$ -free graph G is well-covered. For each induced complete bipartite unbalanced subgraph B of G on vertex sets of bipartition B_X and B_Y , decide polynomially whether B is generating. Once an unbalanced generating subgraph is discovered, the algorithm terminates announcing G is not well-covered. If the algorithm checked all induced complete bipartite unbalanced subgraphs of G , and none of them is generating, then G is well-covered. Hence, the **WC** problem is polynomial when its input is restricted to $K_{1,4}$ -free graphs, but that contradicts Theorem 12. Thus the **GS** problem is **NP**-complete, when its input is a $K_{1,4}$ -free graph. ■

4 Polynomial algorithms when Δ is bounded

Define $\Delta_G = \max\{|N(v)| : v \in V(G)\}$. The main results of this section are polynomial algorithms for the **RE** problem and the **GS** problem in the restricted case that Δ_G is bounded. First, we prove that the **GS** problem is polynomial, when Δ_G is bounded.

Theorem 14 *Let $k \in \mathbb{N}$. The following problem can be solved in $O(n^{2+2k^3})$ time.*

Input: A graph G such that $\Delta_G \leq k \cdot (\log_2 n)^{\frac{1}{3}}$, and an induced complete bipartite subgraph B of G .

Question: Is B generating?

Proof. Let B be an induced complete bipartite subgraph of G on vertex sets of bipartition B_X and B_Y . For every $V \in \{X, Y\}$, let $U \in \{X, Y\} - \{V\}$, and define:

$$M(B_V) = N(B_V) \cap N_2(B_U), M_2(B_V) = N_2(B_V) \cap N_3(B_U).$$

Then $|M(B_V)| \leq k^2(\log_2 n)^{2/3}$ and $|M_2(B_V)| \leq k^3 \log_2 n$. Obviously, B is generating if and only if there exists an independent set in $M_2(B_X) \cup M_2(B_Y)$ that dominates $M(B_X) \cup M(B_Y)$.

The following algorithm decides whether B is generating. For each subset S of $M_2(B_X) \cup M_2(B_Y)$, check whether S is independent and dominates $M(B_X) \cup M(B_Y)$. Once an independent set $S \subseteq M_2(B_X) \cup M_2(B_Y)$ is found such that $M(B_X) \cup M(B_Y) \subseteq N[S]$, the algorithm terminates announcing the instance at hand is positive. If all subsets of $M_2(B_X) \cup M_2(B_Y)$ were checked, and none of them is independent and dominates $M(B_X) \cup M(B_Y)$, then the algorithm returns a negative answer.

The number of subsets the algorithm checks is

$$O(2^{|M_2(B_X) \cup M_2(B_Y)|}) = O(2^{2k^3 \log_2 n}) = O(n^{2k^3}).$$

For each subset S , the decision whether S is both independent and dominates $M(B_X) \cup M(B_Y)$ can be done in $O(n^2)$. Therefore, the algorithm terminates in $O(n^{2+2k^3})$ time, which is polynomial. ■

Theorem 15 [2] *Let $k \in N$. The following problem is polynomial.*

Input: A graph G with $\Delta_G \leq k \cdot (\log_2 n)^{\frac{1}{3}}$, and a function $w : V(G) \rightarrow \mathbb{R}$.

Question: Is G w -well-covered?

We use Theorem 14 to prove a stronger result.

Theorem 16 *Let $k \in N$. The following problem can be solved in $O(n^{3+2k^2+2k^3})$ time.*

Input: A graph G such that $\Delta_G \leq k \cdot (\log_2 n)^{\frac{1}{3}}$.

Output: The vector space $WCW(G)$.

Proof. Let G be a graph such that $\Delta_G \leq k \cdot (\log_2 n)^{\frac{1}{3}}$. For every vertex $v \in V$, let L_v be the vector space of all weight functions $w : V(G) \rightarrow \mathbb{R}$ which satisfy all restrictions of all generating subgraphs which contain the vertex v . Clearly, $WCW(G) = \bigcap_{v \in V(G)} L_v$. Hence, we first present an algorithm for finding L_v for every $v \in V$.

Let $v \in V$. Since the diameter of every complete bipartite graph is at most 2, every complete bipartite subgraph of G which contains v is a subgraph of $N_2[v]$. However,

$$|N_2(v)| \leq \Delta_G^2 \leq k^2(\log_2 n)^{\frac{2}{3}},$$

and

$$|N_2[v]| \leq 2|N_2(v)| \leq 2k^2(\log_2 n)^{\frac{2}{3}}.$$

Therefore, the number of induced complete bipartite subgraphs which contain v can not exceed

$$2^{2k^2(\log_2 n)^{\frac{2}{3}}} \leq n^{2k^2}.$$

The following algorithm finds L_v :

- For each induced complete bipartite subgraph $B = (B_X, B_Y)$ of G containing v :
 - Decide whether B is generating;
 - If B is generating add the restriction $w(B_X) = w(B_Y)$ to the list of equations defining L_v .

We have proved that the number of induced complete bipartite subgraphs of G containing v can not exceed n^{2k^2} . By Theorem 14, deciding for each subgraph whether it is generating can be done in $O(n^{2+2k^3})$ time. Therefore, the algorithm for finding L_v terminates in $O(n^{2+2k^2+2k^3})$ time. In order to find $WCW(G)$, the algorithm for finding L_v should be invoked n times. Therefore, finding $WCW(G)$ can be completed in $O(n^{3+2k^2+2k^3})$ time. ■

We next prove that the **RE** problem is polynomial, when Δ_G is bounded. Note that the bound on Δ_G in the next theorem is different from the bound on Δ_G in Theorem 14.

Theorem 17 *Let $k \in N$. The following problem can be solved in $O(n^{2+2k^2})$ time.*

Input: A graph G such that $\Delta_G \leq k \cdot (\log_2 n)^{\frac{1}{2}}$, and an edge $xy \in E$.

Question: Is xy relating?

Proof. For every $v \in \{x, y\}$, let $u \in \{x, y\} - \{v\}$, and define: $M(v) = N(v) \cap N_2(u)$, $M_2(v) = N_2(v) \cap N_3(u)$. Then $|M(v)| \leq k \cdot (\log_2 n)^{\frac{1}{2}}$ and $|M_2(v)| \leq k^2 \log_2 n$. Clearly, xy is relating if and only if there exists an independent set in $M_2(x) \cup M_2(y)$, which dominates $M(x) \cup M(y)$.

The following algorithm decides whether xy is relating. For each subset S of $M_2(x) \cup M_2(y)$, check whether S is independent and dominates $M(x) \cup M(y)$. Once an independent set $S \subseteq M_2(x) \cup M_2(y)$ is found such that $M(x) \cup M(y) \subseteq N[S]$, the algorithm terminates announcing the instance at hand is positive. If all subsets of $M_2(x) \cup M_2(y)$ were checked, and none of them is both independent and dominates $M(x) \cup M(y)$, then the algorithm returns a negative answer.

The number of subsets the algorithm checks is

$$O(2^{|M_2(x) \cup M_2(y)|}) = O(2^{2k^2 \log_2 n}) = O(n^{2k^2}).$$

For each subset S , the decision whether S is both independent and dominates $M(x) \cup M(y)$ can be done in $O(n^2)$ time. Therefore, the algorithm terminates in $O(n^{2+2k^2})$ time. ■

5 Conclusions and future work

The following table presents complexity results concerning the four major problems presented in this paper. The empty table cells correspond to unsolved cases.

In addition, we are interested in finding some polynomial relaxations of the bipartite case, if any. For instance, is it **NP**-complete to recognize well-covered graphs belonging to $\mathcal{G}(\widehat{C}_3, \widehat{C}_5)$?

Let us emphasize that we do not know whether there exists a family of graphs for which the **RE** problem is polynomial but the **GS** problem is **NP**-complete.

Another interesting open question is whether there exists a family of graphs for which the **GS** problem is polynomial and its corresponding **WCW** problem is **co-NP**-hard. In such a family, if exists, the decision for every induced subgraph whether it is generating or not is polynomial, but finding the vector space $WCW(G)$ obtained from the restrictions of all generating subgraphs is **co-NP**-hard.

Input	WC	WCW	RE	GS
general	co-NPC [4, 18]	co-NPH [4, 18]	NPC [1]	NPC [1]
$K_{1,3}$ -free	P [19]	P [13]	P [20]	P [20]
$K_{1,4}$ -free	co-NPC [3]	co-NPH [3]		NPC this paper
$\mathcal{G}(\widehat{C}_4, \widehat{C}_5)$	P [7]	co-NPH this paper	NPC [12]	NPC [12]
$\mathcal{G}(\widehat{C}_4, \widehat{C}_6)$			P [12]	
$\mathcal{G}(\widehat{C}_5, \widehat{C}_6)$			P [11]	
$\mathcal{G}(\widehat{C}_5, \widehat{C}_6, \widehat{C}_7)$			P [11]	P [11]
$\mathcal{G}(\widehat{C}_4, \widehat{C}_5, \widehat{C}_6)$	P [7]	P [11]	P [11]	P [11]
$\mathcal{G}(\widehat{C}_4, \widehat{C}_6, \widehat{C}_7)$			P [10]	P [10]
bipartite	P [5, 15, 17]	co-NPH this paper	NPC this paper	NPC this paper
$\mathcal{G}(\widehat{C}_3, \widehat{C}_4)$	P [6]	co-NPH this paper		NPC this paper
$\mathcal{G}(\widehat{C}_3, \widehat{C}_4, \widehat{C}_5)$	P [6]	co-NPH this paper		NPC this paper
$\Delta \leq k(\log_2 n)^{\frac{1}{3}}$	P [2]	P this paper	P this paper	P this paper
$\Delta \leq k(\log_2 n)^{\frac{1}{2}}$			P this paper	

References

- [1] J. I. Brown, R. J. Nowakowski, I. E. Zverovich, *The structure of well-covered graphs with no cycles of length 4*, Discrete Mathematics **307** (2007) 2235-2245.
- [2] Y. Caro, N. Ellingham, G. F. Ramey, *Local structure when all maximal independent sets have equal weight*, SIAM Journal on Discrete Mathematics **11** (1998) 644-654.
- [3] Y. Caro, A. Sebő, M. Tarsi, *Recognizing greedy structures*, Journal of Algorithms **20** (1996) 137-156.
- [4] V. Chvatal, P. J. Slater, *A note on well-covered graphs*, Quo vadis, Graph Theory Annals of Discrete Mathematics **55**, North Holland, Amsterdam (1993) 179-182.
- [5] O. Favaron, *Very well covered graphs*, Discrete Mathematics **42** (1982) 177-187.
- [6] A. Finbow, B. Hartnell, R. Nowakowski, *A characterization of well-covered graphs of girth 5 or greater*, Journal of Combinatorial Theory B **57** (1993) 44-68.
- [7] A. Finbow, B. Hartnell, R. Nowakowski, *A characterization of well-covered graphs that contain neither 4- nor 5-cycles*, Journal of Graph Theory **18** (1994) 713-721.
- [8] M. Lesk, M. D. Plummer, W. R. Pulleyblank, *Equimatchable graphs*, Graphs Theory and Combinatorics, B. Bollobas ed., Academic press, London (1984) 239-254.
- [9] V. E. Levit, D. Tankus, *On relating edges in well-covered graphs without cycles of length 4 and 6*, Graph Theory, Computational Intelligence and Thought: Essays Dedicated to Martin Charles Golumbic on the Occasion of His 60th Birthday, Lecture Notes in Computer Science **5420** (2009) 144-147.
- [10] V. E. Levit, D. Tankus *Weighted well-covered graphs without C_4 , C_5 , C_6 , C_7* , Discrete Applied Mathematics **159** (2011) 354-359.
- [11] V. E. Levit, D. Tankus, *Weighted well-covered graphs without cycles of lengths 4, 5 and 6*, arXiv:1210.6918 [cs.DM], <http://arxiv.org/pdf/1210.6918v1.pdf> (available online).
- [12] V. E. Levit, D. Tankus, *On relating edges in graphs without cycles of length 4*, Journal of Discrete Algorithms (2013), <http://dx.doi.org/10.1016/j.jda.2013.09.007>, (available online).

- [13] V. E. Levit, D. Tankus, *Weighted well-covered claw-free graphs*, arXiv:1312.7563 [cs.DM], <http://arxiv.org/pdf/1312.7563v1.pdf> (available online).
- [14] M. D. Plummer, *Some covering concepts in graphs*, Journal of Combinatorial Theory **8** (1970) 91-98.
- [15] M. D. Plummer, *Well-covered graphs: a survey*, Quaestiones Mathematicae **16** (1993) 253-287.
- [16] E. Prisner, J. Topp and P. D. Vestergaard, *Well-covered simplicial, chordal and circular arc graphs*, Journal of Graph Theory **21** (1996) 113-119.
- [17] G. Ravindra, *Well-covered graphs*, Journal of Combinatorics, Information and System Sciences **2** (1977) 20-21.
- [18] R. S. Sankaranarayana, L. K. Stewart, *Complexity results for well-covered graphs*, Networks **22** (1992) 247-262.
- [19] D. Tankus, M. Tarsi, *Well-covered claw-free graphs*, Journal of Combinatorial Theory B **66** (1996) 293-302.
- [20] D. Tankus, M. Tarsi, *The structure of well-covered graphs and the complexity of their recognition problems*, Journal of Combinatorial Theory B **69** (1997) 230-233.
- [21] D. Tankus, M. Tarsi, *Greedily constructing Hamiltonian paths, Hamiltonian cycles and maximum linear forests*, Discrete Mathematics **307** (2007) 1833-1843.